Package ‘eudysbiome’

January 6, 2026

Type Package

Title Cartesian plot and contingency test on 16S Microbial data

Version 1.41.0

Date 2016-05-19

Author Xiaoyuan Zhou, Christine Nardini

Maintainer Xiaoyuan Zhou <zhouxiaoyuan@picb.ac.cn>

Description eudysbiome a package that permits to annotate the differential genera as harmful/harmless based on their ability to contribute to host diseases (as indicated in literature) or unknown based on their ambiguous genus classification. Further, the package statistically measures the eubiotic (harmless genera increase or harmful genera decrease) or dysbiotic(harmless genera decrease or harmful genera increase) impact of a given treatment or environmental change on the (gut-intestinal, GI) microbiome in comparison to the microbiome of the reference condition.

Depends R (>= 3.1.0)

Imports plyr, Rsamtools, R.utils, Biostrings

License GPL-2

LazyData TRUE

RoxygenNote 5.0.1

git_url https://git.bioconductor.org/packages/eudysbiome

git_branch devel

git_last_commit 112407c

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2026-01-06

Contents

assignTax 2
Cartesian 3

contingencyCount 5
contingencyTest 6
diffGenera 7
eudysbiome 7
harmGenera 7
microAnnotate 8
microCount 9
microDiff 9
tableSpecies 10

Index **11**

`assignTax` *Taxonomic Classification*

Description

Assign taxonomic paths to unclassified SSU rRNA sequences, by executing `classify.seqs` in Mothur with the 'Wang' approach.

Usage

```
assignTax(fasta, template = NULL, taxonomy = NULL, ksize = 8, iters = 100, cutoff = 80,
processors = 1, dir.out = "assignTax_out")
```

Arguments

`fasta` a fasta file of rRNA sequences to be assigned with taxonomies, e.g. a set of sequences picked as the representatives of OTUs.

`template` a faste file of rRNA reference sequences, default to download "Silva_119_provisional_release.zip" under "qiime" directory from SILVA archive under , and extract "Silva_119_rep_set97.fna", a representative set of SILVA rRNA references of version119 at 97% sequence identity.

`taxonomy` a taxonomic path file mapping to the template file, default to load "Silva_119_rep_set97.fna" matched "taxonomy" stored in the package.

`ksize`, `iters`, `cutoff`, `processors` parameters used in Classify.seqs by Mothur. `ksize`, kmer size which is a search option with the 'Wang' method and by default to 8. `iters`, iterations by default 100 to calculate the bootstrap confidence score for the assigned taxonomy. `cutoff`, a bootstrap confidence score for the taxonomy assignment, by default 80, which means a minimum 80% sequences were assigned by the same taxonomy, a higher value gives a more strict taxonomy assignment. `processors`, the number of central processing units you use to run the command, by default to 1.

`dir.out` a directory where the assigned files were outputted, by default to create `assignTax_out` directory and output assigned files under this directory.

Details

This function performs 'classify.seqs' by running Mothur in command line mode, hence the executable Mothur on your computer is needed. For unix users, the absolute path of Mothur should be added to the PATH environmental variable and exported. For Windows users, the executable Mothur with extension .exe is required under your disks.

Value

two files under `dir.out`, a `*.taxonomy` file which contains a taxonomic path for each sequence and a `*.tax.summary` file which contains a taxonomic outline indicating the number of sequences that were found at each level (kingdom to species). a list containing the following components: exitStatus an error code ('0' for success) given by the execution of the system Mothur commands, see `system`. stderr, stdout standard errors and outputs by executing Mothur command 'classify.seqs'.

Cartesian *2-D Cartesian plane Plots*

Description

This function plots a Cartesian plane of genus abundance difference across the tested conditions (y-axis) and their harmful/harmless nature (negative/positive x-axis), giving rise to up-utmost right and bottom-utmost left quadrants of microbial eubiotic impact and bottom-utmost right and up-utmost left quadrants of dysbiotic impact.

Usage

```
Cartesian(x,log2 = TRUE,micro.anno = NULL, comp.anno = NULL,
             pch = 16, point.col = NULL,point.alpha = 0.6,ylim = NULL,
             xlab = NULL,ylab = NULL,vlty = 2, hlty = 1, srt = 60,
             font = 3, adj = c(1,1), xaxis = NULL, yaxis = NULL, legend = TRUE,
             box = TRUE,box.col = c("darkblue","yellow"),
             ...)
```

Arguments

`x`	a data frame or numeric matrix of microbial abundance variations from which the plot is produced. Rows indicate the differential microbes, columns indicate the pair-wise conditions. `x` values can either be difference values or be *log2* converted, specified with `log2` parameter.
`log2`	logical, specifying if x values should be *log2* converted; default to `TRUE`.
`micro.anno`	a character vector to annotate all row microbes in `x`; e.g. `"harmless","harmful"`, should be in same length with the microbes. It can be given by the output of `microAnnotate`
`comp.anno`	a character vector of conditions pre-defined from the column pair-wise comparisons, should be in same length with the comparisons; default to the pair-wise comparisons.

pch	a vector of point types. Graphical parameters:see par.
point.col	a vector of colors for the points.
point.alpha	alpha value for points; see adjustcolor.
ylim	limits for the y axis.
xlab	a title for the x axis.
ylab	a title for the y axis.
vlty, hlty	types of vertical and horizontal lines to divide the plane with x-axis and y-axis, respectively.
srt, font, adj	graphical parameters for the text on x-axis, see par.
xaxis	a character or expression vector specifying the labels of x axis by *text*; default to row names of x.
yaxis	a character or expression vector specifying the labels of y axis by *axis*; default to at values in axis.
legend	logical, specifying if the legend should be added to the plot; default to TRUE.
box	logical, specifying if the quadrants should be highlighted by boxes; default to TRUE.
box.col	a vector of colors for the up-utmost right, bottom-utmost left quadrants and bottom-utmost right, up-utmost left quadrants, respectively; default to "dark-blue" and "yellow". If only one color is specified, the other one can be NA.
...	additional parameters passed to the default method, or by it to plot.window, text,mtext,axis, and title to control the appearance of the plot.

Value

The Cartesian plane plot

Examples

```
data(microDiff)
attach(microDiff)

newpar = par()
par(mar = c(6,5.1,4.1,6))
Cartesian(x = data,log2 = TRUE,micro.anno = micro.anno,pch = 16,
         comp.anno = comp.anno,point.col = c("blue","purple","orange"))

par(newpar)
detach(microDiff)
```

contingencyCount *Contingency Table Construction*

Description

Computes the frequencies of the contingency table as the cumulated microbial abundance difference classified into each condition and eubiotic/dysbiotic impact term for examining the significance of the association (contingency) between conditions and impacts by `contingencyTest`.

Usage

```
contingencyCount(x, micro.anno=NULL, comp.anno=NULL)
```

Arguments

`x`	See `x` in `Cartesian`, the `x` values should be difference values without log converted.
`micro.anno`	See `micro.anno` in `Cartesian`.
`comp.anno`	See `comp.anno` in `Cartesian`.

Details

Eubiotic impact is measured by variations of increased harmless and decreased harmful microbes, while the dysbiotic impact is measured by the decreased harmless and increased harmful microbes.

Value

The frequencies of condition-impact terms in contingency table

Examples

```
data(microDiff)
attach(microDiff)

microCount = contingencyCount(x = data, micro.anno = micro.anno,
                      comp.anno = comp.anno)

detach(microDiff)
```

contingencyTest *Contingency test for count data*

Description

Performs Chi-squared test or Fisher's exact test for testing the significance of association between conditions and eubiotic/dysbiotic impacts in a contingency table.

Usage

```
contingencyTest(microCount, chisq = TRUE, fisher = TRUE,
                alternative = c("greater"))
```

Arguments

`microCount`	a *m by 2* data frame or numeric matrix of contingency table with frequencies under each condition-impact term; could be produced from `contingencyCount`.
`chisq`, `fisher`	logical indicating if the Chi-squared test or Fisher's exact test should be performed.
`alternative`	parameter specifying for alternative hypothesis, only used when `fisher` is `TRUE`; see `fisher.test`.

Details

Chi-squared test for testing the probability that the proportions of eubiotic frequencies are different between two conditions; furtherly, the Fisher's exact test for testing whether one condition is more likely to be associated to eubiotic impact. More details, refer to `chisq.test` and `fisher.test`

Value

A list with following components: Chisq Chi-squared test results for each pair-wise condition. Chisq.p the p-values of the Chi-squared tests for all pair-wise conditions. Fisher Fisher's exact test results for each pair-wise condition. Fisher.p the p-values of the Fisher's exact tests for all pair-wise conditions.

See Also

`contingencyCount`, `fisher.test`, `chisq.test`

Examples

```
data(microCount)

test = contingencyTest(microCount,chisq = TRUE,fisher = TRUE,
           alternative = "greater")
chisq.p = test[["Chisq.p"]]
fisher.p = test[["Fisher.p"]]
```

`diffGenera` *Differential microbes in Genus-Species table A data frame containing 10 differential genera and the species included, which was to be annotated as* `"harmful"` *or* `"harmless"`.

Description

Differential microbes in Genus-Species table A data frame containing 10 differential genera and the species included, which was to be annotated as `"harmful"` or `"harmless"`.

Usage

```
data(diffGenera)
```

Format

A data frame with 26 rows and 2 columns specifying for `Genus` and `Species`.

`eudysbiome` *eudysbiome.*

Description

eudysbiome.

`harmGenera` *Manually curated genera annotation table A data frame containing 260 genera annotated as "harmful" and the harmful species included in these genera.*

Description

Manually curated genera annotation table A data frame containing 260 genera annotated as "harmful" and the harmful species included in these genera.

Usage

```
data(harmGenera)
```

Format

A data frame with 900 rows and 3 columns specifying for `Genus` and `Species` and the references.

microAnnotate *Genus Annotation*

Description

Annotates given genera as harmful or harmless based on either our manually curated, harmful Genus-Species table in data harmGenera of this package or user defined table.

Usage

```
microAnnotate(microbe, species = TRUE, annotated.micro = NULL)
```

Arguments

microbe	a genus list to be annotated; a Genus-Species data frame which represents the genera and the included corresponding species is recommended to be provided by users for the more accurate annotations, see tableSpecies.
species	logical, specifying if the species are provided in the microbe for the annotations; default to TRUE.
annotated.micro	the annotated genera which are used for the annotation of microbe, it could either be loaded from the data harmGenera or defined by users.

Value

The annotated genera.

Examples

```
#load the genera to be annotated
library(eudysbiome)
data(diffGenera)

#load the curated Genus-Species annotation table
data(harmGenera)

microAnnotate(microbe = diffGenera, species = TRUE,
           annotated.micro = harmGenera)
```

microCount — *Microbial count contingency table*

Description

A matrix containing the counts of differential microbe classified into each condition-eubiotic/dysbiotic impact couple. Rows represent the condition comparisons, columns represent the eubiotic and dysbiotic impacts:

Usage

```
data(microCount)
```

Format

A data frame with 2 rows and 2 variables

Details

- EI. eubiotic impact
- DI. dysbiotic impact

The table can be produced by `microCount` function.

microDiff — *Differential annotated genera with abundance variations among pairwise condition comparisons*

Description

A list containing: i) a data frame of 10 differentila genera with abundance differences among 3 condition comparisons, in which row represents the differential microbes and column represents the comparisons; ii) Genera annotations for the 10 differential genera; iii) pre-defined condition comparison names

Usage

```
data(microDiff)
```

Format

A list

tableSpecies — *Construct a Genus-Species Data Frame*

Description

This function is used to extract only Genus-Species data from the assigned taxonomic paths outputted by `assignTax` and construct a table containing the classified genera and species included in these genera correspondingly.

Usage

```
tableSpecies(tax.file, microbe)
```

Arguments

`tax.file`	a taxonomy file with SSU rRNA sequence names and assigned taxonomic paths, see "*.taxonomy" file outputted by `assignTax`.
`microbe`	a character vector specifying the genera used to construct the Genus-Species data frame

Details

The outputted Genus-Species table can be used as input for the more accurate genus annotation, which annotates genera as `harmful` or `harmless` based on their ability to contribute to mammals' host diseases by `microAnnotate`.

Value

a Genus-Species data frame, only with the genera specified by `microbe` and the included corresponding species .

Examples

```
#a table with "Lactobacillus" and "Bacteroids" genera and the included species
genera = c("Lactobacillus","Bacteroides")
#not excute
#tableSpecies(tax.file = "test.taxExtract.wang.taxonomy", microbe = genera)
```

Index

∗ **datasets**
    diffGenera, 7
    harmGenera, 7
    microCount, 9
    microDiff, 9

adjustcolor, *4*
assignTax, 2, *10*
axis, *4*

Cartesian, 3, *5*
chisq.test, *6*
contingencyCount, 5, *6*
contingencyTest, *5*, 6

diffGenera, 7

eudysbiome, 7
eudysbiome-package *(*eudysbiome*)*, 7

fisher.test, *6*

harmGenera, 7

microAnnotate, *3*, 8, *10*
microCount, *9*, 9
microDiff, 9
mtext, *4*

par, *4*
plot.window, *4*

system, *3*

tableSpecies, *8*, 10
text, *4*
title, *4*